

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
ИНСТИТУТ КОСМИЧЕСКИХ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ  
Кафедра «Информатика»

УТВЕРЖДАЮ  
Заведующий кафедрой  
\_\_\_\_\_ А. С. Кузнецов

« \_\_\_\_ » \_\_\_\_\_ 2017 г.

**БАКАЛАВРСКАЯ РАБОТА**

27.03.03 «Системный анализ и управление»

Разработка экспертного модуля по сборке изделия из комплектующих

Руководитель \_\_\_\_\_ доцент, канд. техн. наук А.А. Даничев  
подпись, дата

Выпускник \_\_\_\_\_ Е.С. Кемеров  
подпись, дата

Красноярск 2017

## **РЕФЕРАТ**

Выпускная квалификационная работа по теме «Разработка экспертного модуля по сборке изделия из комплектующих» содержит 49 страниц текстового документа, 17 использованных источников, 21 иллюстрацию.

**ПРОГРАММНЫЙ МОДУЛЬ, СБОРКА ПК, МЕТОД ВЕТВЕЙ И ГРАНИЦ,  
СБОРКА ИЗДЕЛИЯ ИЗ КОМПЛЕКТУЮЩИХ.**

Целью работы является разработка программного модуля по сборке персонального компьютера из комплектующих с возможностью использования сторонними системами, а также адаптации модуля под другую предметную область.

В ходе выполнения данной работы был проведен анализ предметной области и существующих систем по подбору ПК, реализован синтаксический анализатор, сформирована база данных и разработаны алгоритмы работы программной системы, создан пользовательский интерфейс программного модуля.

В результате был реализован программный модуль, осуществляющий сборку изделия из комплектующих по заданным критериям.

## СОДЕРЖАНИЕ

Введение .....	4
1 Анализ предметной области .....	6
1.1 Понятие персонального компьютера.....	6
1.2 Основные комплектующие персонального компьютера.....	7
1.3 Анализ существующих систем по подбору ПК.....	19
1.4 Метод ветвей и границ.....	24
1.4.1 Формальное описание.....	25
1.5 Базы данных и базы знаний .....	28
2 Разработка программного модуля по сборке ПК из комплектующих.....	33
2.1 Модель программной системы по сборке ПК из комплектующих .....	33
2.2 Средства разработки программного модуля.....	34
2.2.1 PostgreSQL 9.5 .....	34
2.2.1.1 Руководство по настройке.....	34
2.2.1.2 Требования по настройке .....	35
2.2.2 Django 1.11 .....	35
2.2.2.1 Краткая характеристика .....	35
2.2.2.2 Руководство по настройке.....	35
2.2.3 JetBrains PyCharm.....	36
2.2.3.1 Краткая характеристика .....	36
2.2.3.2 Руководство по настройке.....	36
2.2.3.3 Требования по настройке .....	36
2.3 Формирование базы данных.....	36
2.4 Разработка алгоритмов работы программной системы .....	42
Заключение .....	49
Список использованных источников .....	50

## **ВВЕДЕНИЕ**

Жизнь современного человека тесно связана с различными техническими устройствами и в первую очередь с компьютерами. Персональные компьютеры уже давно и прочно вошли в наш повседневный обиход. Сфера их применения необычайно широка: образование, медицина, производство, банковские услуги, киноиндустрия – во всех этих направлениях с успехом пользуются неограниченными возможностями компьютерной техники. Сложно представить современное рабочее место без нее. Компьютеры позволяют оптимизировать и автоматизировать различные рабочие процессы, хранить большие объемы информации и оперативно получать к ней доступ, удаленно управлять производственными мощностями, ускорить и упростить взаимодействие между специалистами, проводить сложные технические расчеты и вычисления и многое другое.

Но компьютеры прочно утвердились не только в рабочей сфере, они стали неотъемлемой частью и повседневной жизни человека. Вместе с интернетом компьютер дает широчайшие возможности для отдыха и досуга.

В связи с этим перед людьми встает важный и непростой вопрос выбора подходящей компьютерной техники. Осложняется он, в первую очередь, необходимостью обладать специальными знаниями о внутреннем устройстве и функционировании ПК. Вторым, не менее значимым аспектом, затрудняющим процесс выбора является быстрая и непрерывная смена технологий. Поэтому проблема подбора подходящего компьютера и модернизации техники будет актуальна на протяжении еще многих лет.

Здесь на помощь человеку может прийти система по сборке изделий из комплектующих. Такая система представляет собой программное средство, использующее базу знаний для обеспечения высокоэффективного решения неформализованных задач в узкой предметной области.

Объектом исследования данной работы являются программные системы по сборке изделий из комплектующих.

Предметом исследования работы является программная система по сборке ПК из комплектующих, отвечающего требованиям пользователя.

Целью работы является разработка программного модуля по сборке персонального компьютера из комплектующих с возможностью использования сторонними системами, а также адаптации модуля под другую предметную область.

Задачи:

- сформулировать определение понятия «персональный компьютер» в рамках данной работы;
- определить состав и характеристики комплектующих для сборки персонального компьютера;
- провести анализ существующих систем по подбору ПК;
- рассмотреть метод ветвей и границ;
- изучить понятия базы данных и базы знаний, их сходства и различия;
- составить модель программной системы по сборке ПК из комплектующих;
- сформировать базу данных исходных материалов;
- разработать алгоритмы работы программной системы;
- создать пользовательский интерфейс программного модуля;
- представить процесс работы программного модуля по сборке ПК из комплектующих, его функциональные возможности и сферу применения.

## **1 Анализ предметной области**

### **1.1 Понятие персонального компьютера**

«Персональный компьютер» – популярный и распространенный термин, в современной действительности мы встречаемся с ним ежедневно. Порой под ПК подразумевают практически все виды компьютерных устройств от десктопных решений до мобильных (ноутбук, планшетный компьютер).

Понятие «персональный компьютер» происходит от английских слов – «personal» (личный, персональный) и «computer» (компьютер). Слово компьютер является производным от английских слов to compute, computer, которые переводятся как «вычислять», «вычислитель» (английское слово, в свою очередь, происходит от латинского computāre – «вычислять»). Первое использование термина относилось к компьютеру Programma 101 (1964) итальянской фирмы Olivetti. Возникновение такого названия было связано с тем, что после начала массового производства микросхем, стоимость компьютеров резко упала. Это привело к созданию вместо многопользовательских мэйнфреймов компьютеров, которые эксплуатировались одним человеком. Впоследствии этот термин был перенесён на другие компьютеры.

В Советском Союзе вычислительные машины, предназначенные для персонального использования, носили официальное название «персональные электронные вычислительные машины» (ПЭВМ). В терминологии, принятой в российских стандартах, это словосочетание и сегодня указывается вместо используемого названия «персональный компьютер».

В данной работе в качестве понятия «персональный компьютер» рассматривается следующая дефиниция.

Персональный компьютер – устройство или система, способная выполнять заданную, чётко определённую, изменяемую последовательность операций.

Чтобы различать типологию ПК существуют уточняющие термины: десктоп, ноутбук (лэптоп), нетбук, планшетный компьютер и т.д.

Конструктивно персональный компьютер включает в себя следующие компоненты:

- материнская плата;
- процессор;
- оперативная память;
- жесткий диск;
- блок питания;
- видеокарта;
- монитор;
- система охлаждения;
- звуковая карта;
- устройства ввода (клавиатура, мышь).

## **1.2 Основные комплектующие персонального компьютера**

Как и любое другое достаточно сложное устройство, персональный компьютер состоит из комплектующих, которые в совокупности работают как слаженный механизм. Комплектующие бывают внутренними и внешними. Внутренние комплектующие представляют собой устройства, которые устанавливаются внутри системного блока компьютера, – материнская плата, процессор, оперативная память и т.д. Внешние комплектующие – устройства, которые подключаются к компьютеру любым доступным способом и при этом остаются снаружи системного блока. Примерами таких устройств могут быть внешние жесткие диски, USB-накопители, USB-сетевые карты и т.д. Количество внутренних и внешних устройств, которые можно устанавливать или подключать к персональному компьютеру, с каждым днем растет, а их выбор зависит от технической фантазии человека и его технических потребностей [2]. Основными комплектующими персонального компьютера являются:

- материнская плата;

- процессор;
- оперативная память;
- жесткий диск;
- блок питания;
- видеокарта;
- система охлаждения;
- звуковая карта;
- устройства ввода (клавиатура, мышь);
- монитор.



Рисунок 1 – Схема типичной архитектуры ПК

**Материнская плата** – системная плата, основанная на чипсете (набор микросхем), которая дает возможность общения между процессором, оперативной памятью, видеопамятью и другими устройствами системного блока. В состав чипсета входят два основных чипа – северный (системный контроллер) и южный мост (функциональный контроллер). Основная задача северного моста – обеспечение связи процессора с оперативной памятью и видеосистемой. А основная задача южного моста – обмен данными с северным



мостом и различными периферийными устройствами. Функциональные возможности южного моста – контроллеры жестких дисков, контроллер дисководов, контроллер шин PCI, USB-контроллер, контроллер звука, сетевого интерфейса, контроллеры портов ввода/вывода.



Рисунок 2 – Внешний вид материнской платы

Основными характеристиками материнских плат являются:

- модель системной платы (название);
- набор микросхем (чипсет);
- тип разъема для процессора (socket);
- форм-фактор (размер платы);
- слот оперативной памяти (тип, кол-во);
- разъёмы PCI-Express (кол-во);
- разъемы E-Sata;
- различные разъемы (usb, ieee1394a и др.);

- встроенные звуковая и сетевая карты;
- поддержка RAID-массивов.

**Центральный процессор (Central Processing Unit, CPU)** – это один из основных компонентов компьютера, который выполняет арифметические и логические операции, задаваемые программой.

Физически процессор представляет собой интегральную микросхему, на которой размещены электронные блоки, реализующие все его функции [2].

На сегодняшний день на рынке присутствует только два реальных производителя процессоров, а именно AMD и Intel. Поэтому их и рассматривают, когда речь идет о выборе процессора.

Все процессоры различаются интерфейсом, используемыми технологиями (алгоритмами, количеством ядер) и быстродействием.



Рисунок 3 – Внешний вид процессора AMD – Ryzen 5 1600

Основные характеристики процессоров:

- модель процессора (название);
- производитель ядра (название);

- количество ядер (число);
- тактовая частота ядра (МГц);
- технологический процесс (нм);
- тепловая расчетная мощность (Вт);
- частота системной шины (МГц);
- объем кэш-памяти (1-3 уровни);
- поддержка различных технологий (число);
- габаритные размеры (мм).

**Оперативная память (Random Access Memory, RAM)** – модули памяти, необходимы для работы любой операционной системы, приложений и программ, а также для обработки и временного хранения данных. Оперативная память выполняется по типам, но сейчас самый популярный тип памяти – DDR3. Объем оперативной памяти существенно влияет на общую производительность системы [2].



Рисунок 4 – Внешний вид модуля оперативной памяти

Основные характеристики модуля оперативной памяти:

- Производитель памяти (название)
- Тип модуля памяти (DDR, DDR2, DDR3, DDR4)
- Корпус исполнения (технология)
- Разъем (кол-во pin)
- Напряжение питания (В)
- Тактовая частота памяти (МГц)

- Частота шины (МГц)
- Тайминги памяти (цифры)
- Пропускная способность (Гбайт/с)

**Жесткий диск (Hard Disk Drive, HDD)** – устройство, используемое для постоянного хранения информации, необходимой для работы на компьютере, и доступа к ней. Современные жесткие диски отличаются высокими показателями емкости (несколько ТераБайт), скорости и надежности, а также низкой стоимостью. Жесткие диски имеют свои интерфейсы подключения, такие как: IDE (менее используется), SATA (широкое применение), SCSI (серверные платформы). Основные различия жестких дисков – интерфейс подключения к материнской плате, скорость вращения пластин, объем хранимых данных, кэш-буфер, время позиционирования головок, время поиска информации и др. Кроме того, жесткие диски могут быть внутренними и внешними.



Рисунок 5 – Внешний вид жесткого диска Western Digital – RE3

Основные характеристики жесткого диска:

- марка производителя (название);
- интерфейс подключения (IDE, SATA);
- объем (ГБ);
- кэш-память (МБ);
- скорость вращения (об/мин);
- макс. Скорость передачи данных (МБ/с);
- пост. Скорость передачи данных (МБ/с);
- средняя задержка (мс).

**Блок питания** – блок питания является неотъемлемой частью системного блока, с помощью него подается питание на материнскую плату, жесткие диски и видеокарты. Блоки питания различаются лишь производителем и номинальной мощностью. В среднем эта цифра колеблется от 400 до 600 Ватт. Мощный и стабильный блок питания сегодня является обязательным требованием для персонального компьютера. Если напряжение линий блока питания будет нестабильным (или значительно отличаться от эталонных напряжений), системные компоненты могут выйти из строя. В частности, материнские платы и процессоры очень чувствительны к нестабильному питанию.



Рисунок 6 – Внешний вид блока питания

Основные характеристики блока питания:

- тип блока питания (назв., модель);
- номинальная мощность (Вт);
- кол-во разъемов питания (шт).

**Видеокарта (видеоадаптер)** – представляет собой графическую подсистему компьютера и служит для формирования и вывода на монитор изображения. От него зависят качество изображения и скорость визуализации двух- и трехмерной графики. Наибольшие требования предъявляются именно к последнему пункту, поскольку все современные игры и графические программы для обработки сложных 3D-объектов используют именно 3D-возможности видеоадаптера. Кроме своей прямой обязанности, видеокарта может выполнять некоторые другие функции, например выводить параллельное изображение на телевизионный приемник или осуществлять захват изображения. В этом случае на планке видеокарты, кроме разъемов для подключения мониторов, имеются разъемы для выполнения соответствующих задач [2].





Рисунок 7 – Внешний вид видеокарты Gigabyte – GeForce GTX 1080 Ti

Основные характеристики видеокарты:

- модель видеопроцессора (название);
- объем памяти (Гбайт);
- номинальная частота памяти (МГц);
- частота GPU (МГц);
- частота шейдерных блоков (МГц);
- ширина шины памяти (бит);
- технологический процесс (нм);
- тепловая расчетная мощность (Вт);
- различные разъемы (кол-во);
- поддержка различных технологий (кол-во).

**Монитор** – устройство, служащее для визуального отображения текстовой и графической информации, которую формирует видеоадаптер. Мониторы разделяются по размеру экрана – диагонали. Основные особенности: время отклика, максимальное разрешение в пикселях, время отклика и угол обзора.



Рисунок 8 – Внешний вид монитора Dell UltraSharp U3415W

Основными характеристиками монитора являются:

- марка производителя (название);
- модель монитора (маркировка) ;
- диагональ монитора (дюйм) ;
- максимальное разрешение (пиксели) ;
- динамический контраст (х:х) ;
- время отклика (мс) ;
- разъемы подключения (кол-во, тип) ;
- угол обзора (градусы) ;
- частота кадров (Гц) .

**Система охлаждения** – набор средств для отвода тепла от нагревающихся в процессе работы компьютерных компонентов. Поскольку процессор является одним из самых главных устройств в компьютере, условия функционирования нужно обеспечивать для него соответствующие. И самое важное из них – эффективная система охлаждения. Существует несколько способов охлаждения: воздушное, жидкостное, фреоновая установка, системы открытого испарения.



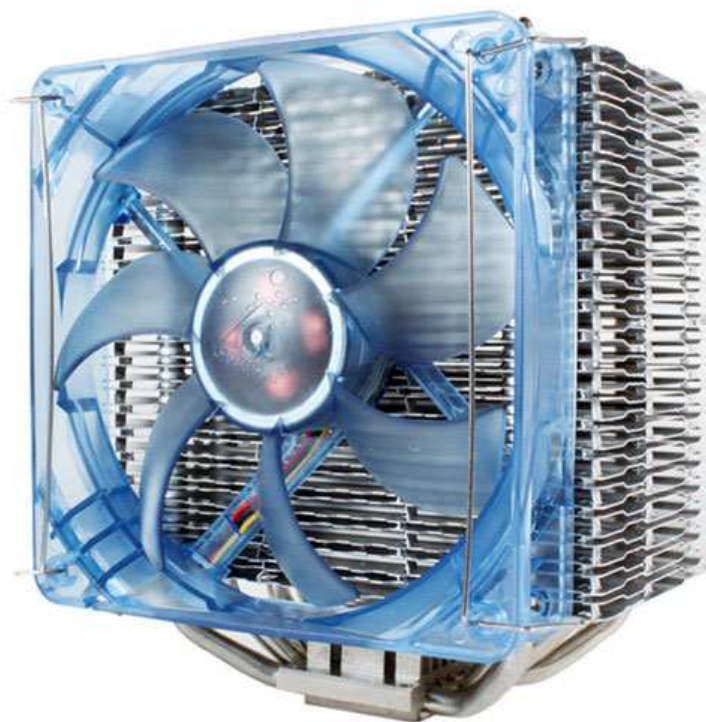


Рисунок 8 – Внешний вид воздушной системы охлаждения

**Звуковая карта (адаптер)** – устройство, в составе которого имеется звуковой процессор и другие вспомогательные компоненты, с помощью которых формируется звуковой сигнал необходимого уровня и окраски. Далее этот сигнал отправляется на акустическую систему, позволяя вам тем самым слышать звук.

Звуковые карты бывают различного исполнения. Чаще всего это интегрированное в материнскую плату решение или карта, устанавливаемая в PCI-слот. Вместе с тем достаточно часто встречаются внешние устройства, представляющие собой профессиональные высококачественные звуковые решения.



Рисунок 9 – Внешний вид звуковой карты

**Устройства ввода** – периферийное оборудование, предназначенное для ввода данных или сигналов в компьютер. Основным и самыми распространенными устройствами ввода являются клавиатура и компьютерная мышь. Эти устройства бывают двух видов – проводными и беспроводными.



Рисунок 10 – Внешний вид клавиатуры и компьютерной мыши

### 1.3 Анализ существующих систем по подбору ПК

Для формирования значимых требований, которым должна отвечать программная система по сборке ПК из комплектующих, был проведен мониторинг интернет-сервисов, предоставляющих данную услугу. В качестве примеров для сравнения были выбраны пять ресурсов:

- <https://pcpartpicker.com>;

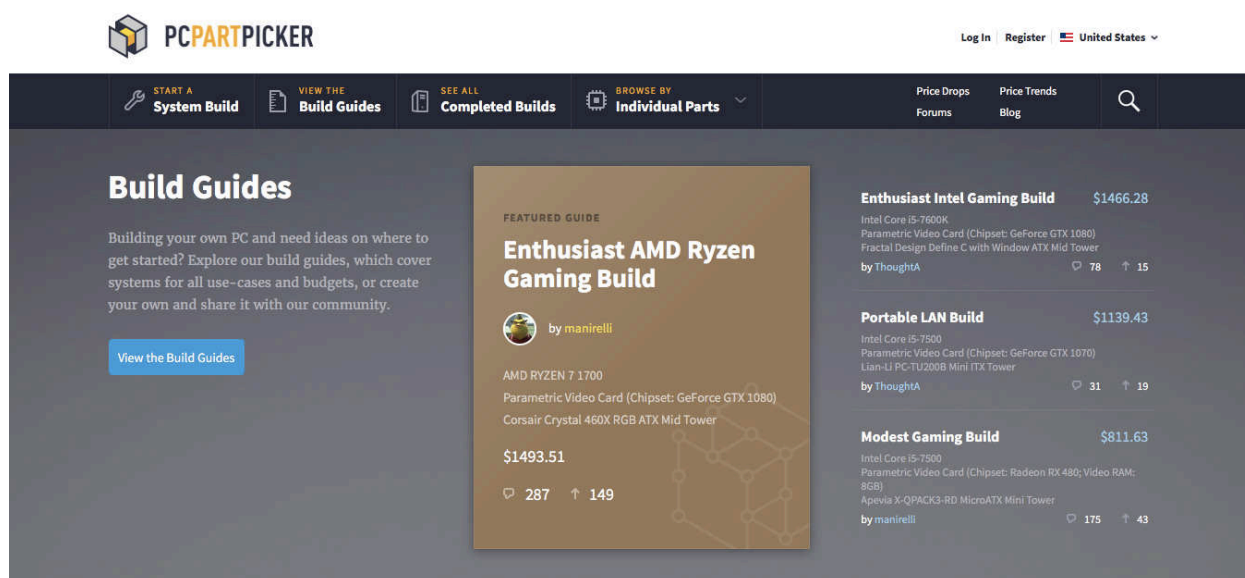


Рисунок 11 – Скриншот главной страницы <https://pcpartpicker.com>

Данный сайт представляет собой сервис готовых примеров сборки ПК, которые формируются пользователями-экспертами. Функционал сайта включает в себя: создание собственных сборок ПК, просмотр готовых решений, поиск по отдельным комплектующим.

Среди преимуществ данного сервиса можно выделить: актуальность сборок ПК, мониторинг цен на комплектующие, подробные инструкции от пользователей, система рейтинга, фильтр совместимости комплектующих.

К недостаткам можно отнести: отсутствие возможности автоматического создания конфигураций, зависимость качества сборки изделия от компетентности пользователя, недоступность для неопытного пользователя, ориентация только на зарубежную аудиторию.

– <http://www.ironbook.ru/constructor>;

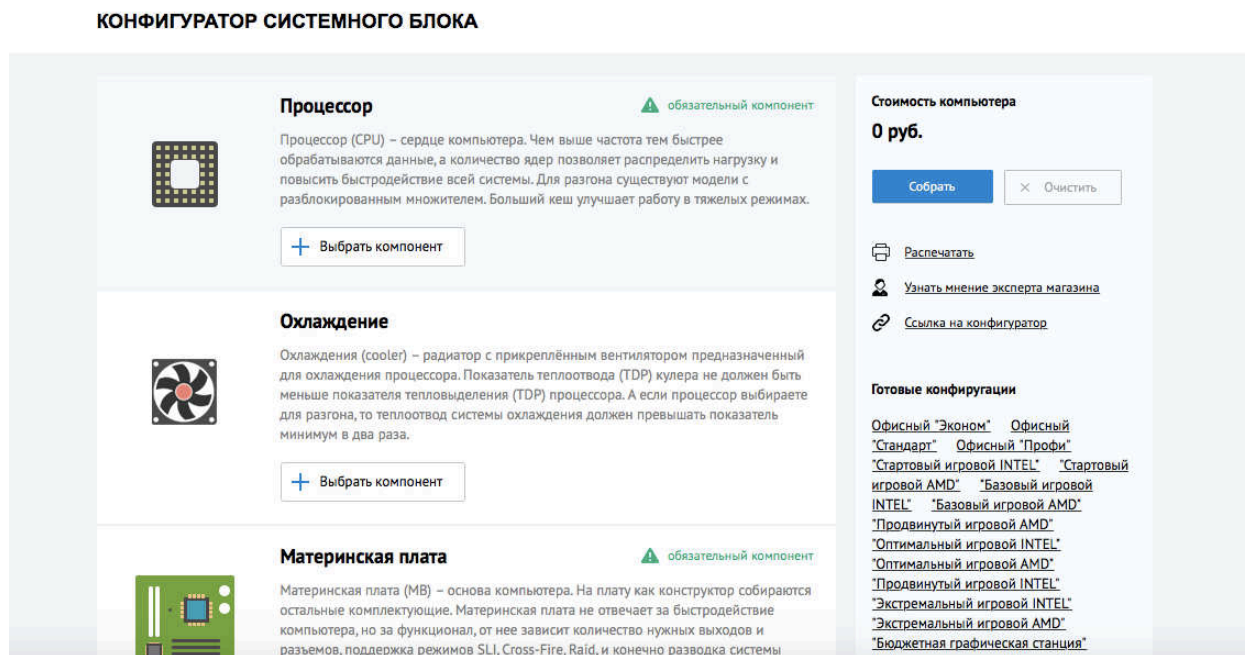


Рисунок 12 – Скриншот конфигуратора системного блока на сайте

<http://www.ironbook.ru/constructor>

Портал «Ironbook» – это интернет-магазин компьютерной техники, в функционал которого встроен конфигуратор системного блока. Он предоставляет пользователю возможность создавать сборки ПК из имеющихся в магазине комплектующих, а также выбрать одну из шестнадцати готовых конфигураций.

К плюсам данного сервиса относятся: фильтр совместимости комплектующих, краткое текстовое пояснение к каждому типу комплектующих, интуитивно-понятный интерфейс.

Минусами можно считать: отсутствие возможности автоматического создания индивидуальных конфигураций, отсутствие возможности подбора монитора и устройств ввода, выбор комплектующих ограничен рамками данного интернет-магазина.

– <http://www.compday.ru/configurator-pc.html>;

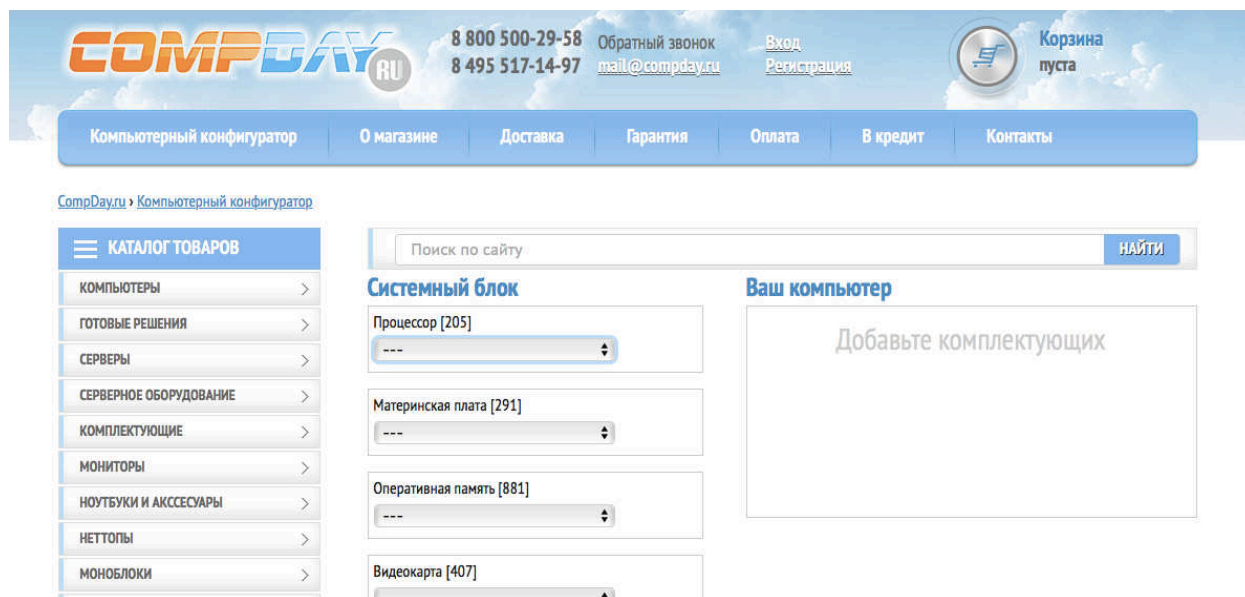


Рисунок 13 – Скриншот компьютерного конфигуратора на сайте  
<http://www.compday.ru/configurator-pc.html>

Компьютерный конфигуратор, представленный на сайте compday.ru, также является составной частью интернет-магазина.

В качестве положительных сторон данного сервиса можно отметить: фильтр совместимости комплектующих, возможность подбора всех типов комплектующих.

К отрицательным сторонам относятся: отсутствие возможности автоматического создания конфигураций, отсутствие готовых вариантов сборок, недоступность для неопытного пользователя, неудобный пользовательский интерфейс, выбор комплектующих ограничен рамками данного интернет-магазина.



– <http://soberisam.com/build>;

Рисунок 14 – Скриншот компьютерного конфигуратора на сайте

<http://soberisam.com/build/>

Сайт [soberisam.com](http://soberisam.com) является интернет-магазином, специализирующимся на сборке ПК из комплектующих и предоставляет пользователям возможность собрать самому компьютер, с помощью встроенного конфигуратора, выбрать один из пятисот вариантов готовых профессиональных сборок, кроме того, на сайте представлены примеры сборок ПК, созданных непосредственно пользователями ресурса.

Из плюсов: большой выбор готовых решений, краткое текстовое пояснение к каждому типу комплектующих, интуитивно-понятный интерфейс, фильтр совместимости комплектующих.

Среди минусов: отсутствие возможности автоматического создания конфигураций, недоступность для неопытного пользователя, отсутствие возможности подбора монитора и устройств ввода, выбор комплектующих ограничен рамками данного интернет-магазина, зависимость качества пользовательской сборки изделия от компетентности пользователя.

– <https://gaming.edelws.ru/constructor>.

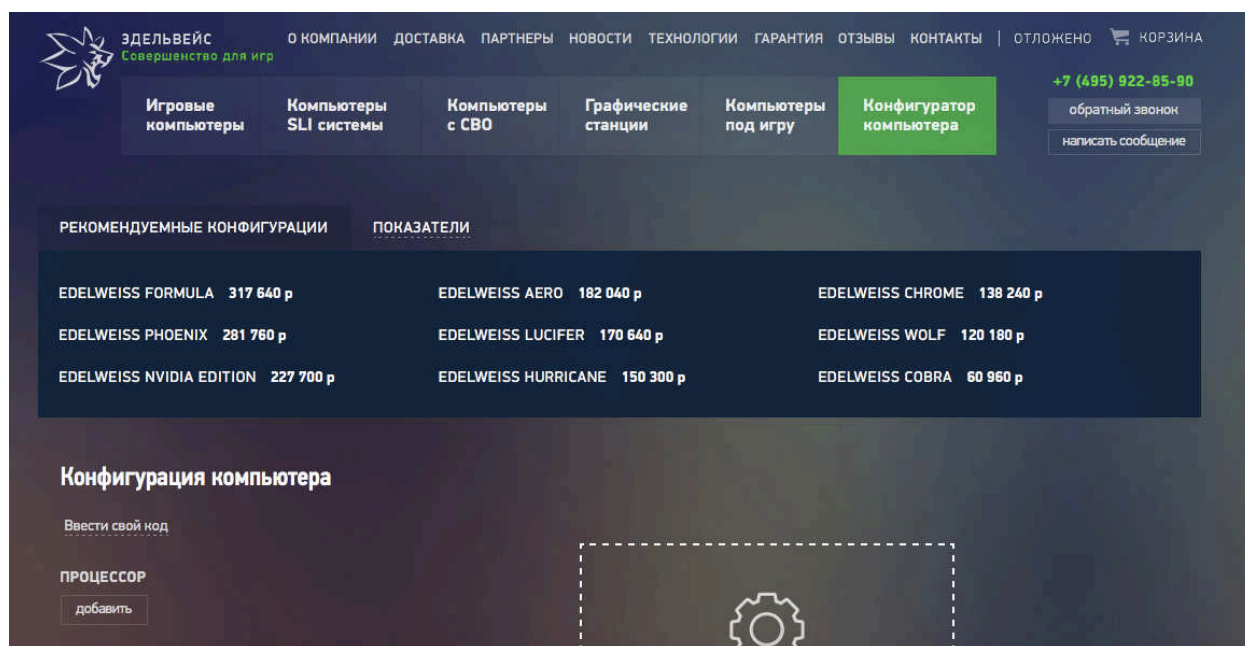


Рисунок 15 – Скриншот компьютерного конфигуратора на сайте

<https://gaming.edelws.ru/constructor>

Компания «Эдельвейс» специализируется на сборке исключительно игровых ПК и имеет собственный конфигуратор сборки компьютера. В качестве готовых конфигураций компания предлагает девять вариантов.

К положительным сторонам можно отнести: краткое текстовое пояснение к каждому комплектующему, интуитивно-понятный интерфейс и фильтр совместимости комплектующих.

Среди отрицательных аспектов: отсутствие возможности автоматического создания конфигураций, недоступность для неопытного пользователя, отсутствие возможности подбора монитора и устройств ввода, выбор комплектующих ограничен рамками данного интернет-магазина, выбор конфигураций ПК ограничен игровым сегментом.

Таким образом, в рамках мониторинга специализированных ресурсов были выделены следующие критерии, которым должен соответствовать разрабатываемый программный модуль:

– возможность создания автоматических конфигураций ПК;

- доступность для неподготовленного пользователя;
- наличие фильтра совместимости комплектующих;
- широкий ассортимент комплектующих ПК;
- возможность выбора полного спектра комплектующих (выбор не только внутренних комплектующих, но и устройств ввода/вывода);
- независимость качества конфигурации от опыта пользователя.

#### **1.4 Метод ветвей и границ**

При рассмотрении задач дискретной оптимизации, стоит отметить, что все они обладают конечным множеством допустимых решений, которые теоретически возможно перебрать и определить наилучшее – дающее максимум или минимум целевой функции. На практике это чаще всего сложно осуществить даже для задач небольшой размерности.

В методах неявного перебора делается попытка так организовать перебор, используя свойства рассматриваемой задачи, чтобы отбросить часть допустимых решений. Наибольшее распространение среди схем неявного перебора получил метод ветвей и границ, в основе которого лежит идея последовательного разбиения множества допустимых решений. На каждом шаге метода элементы разбиения (подмножества) подвергаются анализу – содержит ли данное подмножество оптимальное решение или нет. Если рассматривается задача на минимум, то проверка осуществляется путем сравнения нижней оценки значения целевой функции на данном подмножестве с верхней оценкой функционала. В качестве оценки сверху используется значение целевой функции на некотором допустимом решении. Допустимое решение, дающее наименьшую верхнюю оценку, называют рекордом. Если оценка снизу целевой функции на данном подмножестве не меньше оценки сверху, то рассматриваемое подмножество не содержит решения лучше рекорда и может быть отброшено. Если значение целевой функции на очередном решении меньше рекордного, то



происходит смена рекорда. Будем говорить, что подмножество решений просмотрено, если установлено, что оно не содержит решения лучше рекорда.

Если просмотрены все элементы разбиения, алгоритм завершает работу, а текущий рекорд является оптимальным решением. В противном случае среди непросмотренных элементов разбиения выбирается множество, являющееся в определенном смысле перспективным. Оно подвергается разбиению (ветвлению). Новые подмножества анализируются по описанной выше схеме. Процесс продолжается до тех пор, пока не будут просмотрены все элементы разбиения.

#### 1.4.1 Формальное описание

Пусть рассматривается задача вида

$$f(x) \rightarrow \min_{x \in D}, \quad (1.1)$$

где  $f(x)$  – вещественная функция;

$D$  – конечное множество допустимых решений.

Пусть  $d \subseteq D$ . Функцию  $b(d)$ , ставящую в соответствие множеству  $d$  разбиение его на подмножества  $d_1, \dots, d_N$ ,  $N > 1$ , будем называть ветвлением.

Вещественная функция  $H(d)$  называется нижней границей для  $d$ , если

$$1) \ H(d) \leq \min_{x \in d} f(x); \quad (1.2)$$

2) на одноэлементном множестве  $\{x\}$  верно равенство

$$H(\{x\}) = f(x). \quad (1.3)$$

Алгоритм, реализующий метод ветвей и границ, состоит из последовательности однотипных шагов. На каждом шаге известен рекорд  $x^0$  и подмножества  $t_1, t_2, \dots, t_L$  непросмотренных решений. В начале работы алгоритма  $L = 1, t_1 = D, x^0$  – произвольный элемент множества  $D$  или пустое множество (на пустом множестве положим значение функционала равным бесконечности).

На каждом шаге алгоритм начинает работу с проверки элементов разбиения. Пусть проверяется множество  $t_j$ . Множество  $t_j$  отсекается в одном из двух, последовательно проверяемых случаев:

1) если

$$H(t_j) \geq f(x^0); \quad (1.4)$$

2) если

$$H(t_j) < f(x^0) \quad (1.5)$$

и найден такой элемент  $y_j \in t_j$ , что  $f(y_j) = \min_{x \in t_j} f(x) = H(t_j)$ .

В случае 2 происходит смена рекорда  $x^0 = y_j$ .

Пусть  $t_1, t_2, \dots, t_M$  ( $M \leq L$ ) – неотсеченные множества (будем считать, что отсечены множества с номерами  $M + 1, \dots, L$ ).

В случае  $M = 0$  алгоритм заканчивает работу, и в качестве решения задачи принимается рекорд  $x^0$ . При  $M \geq 1$  среди множеств  $t_1, \dots, t_M$  выбирается множество для нового ветвления. Пусть таковым является множество  $t_1$ . Тогда осуществляется ветвление  $b(t_1) = (d_1, \dots, d_N)$ , в результате которого получаем список множеств  $d_1, \dots, d_N, t_2, \dots, t_M$ . Эти множества нумеруются числами от 1 до  $L$ , и начинается новый шаг алгоритма.

Нетрудно убедиться в том, что описанный алгоритм находит оптимальное решение за конечное число шагов.

Описанная последовательность действий является общей схемой метода ветвей и границ для решения задач на минимум. При решении конкретной задачи следует указать способы построения нижней и верхней оценок, метод ветвления, а также правило выбора перспективного множества для разбиения.

При определении «перспективного» элемента разбиения в основном применяются две схемы: одновременного (многостороннего) и одностороннего ветвления. При одновременном ветвлении функция  $b$  может быть применена к любому элементу разбиения. Часто в качестве такого элемента выбирается подмножество  $t_k$  с минимальной нижней границей

$$H(t_k) = \min_{1 \leq i \leq L} H(t_i). \quad (1.6)$$

При одностороннем ветвлении номер разбиваемого подмножества известен заранее. В этом случае, не ограничивая общности, можно считать, что «перспективным» является подмножество  $t_1$ . Отметим, что при односторонней схеме ветвления нет необходимости запоминать все элементы разбиения, достаточно иметь информацию о первом элементе разбиения и объединении остальных элементов.

Разбиения множеств решений (ветвление) удобно представлять в виде дерева решений. На рисунке 16 приведены примеры одновременной (1) и односторонней (2) схем ветвления. Каждая вершина дерева соответствует некоторому подмножеству решений. Дуги, исходящие из вершины, означают, что на некотором шаге это подмножество отсечь не удалось и оно было разбито на подмножества. Вершины, в которые входят эти дуги, соответствуют подмножествам, полученным в результате ветвления. Зачеркнутые висячие вершины означают отсеченные подмножества. Незачеркнутые висячие вершины соответствуют несмотренным множествам, среди которых на следующем шаге алгоритма выбирается подмножество для дальнейшего ветвления.

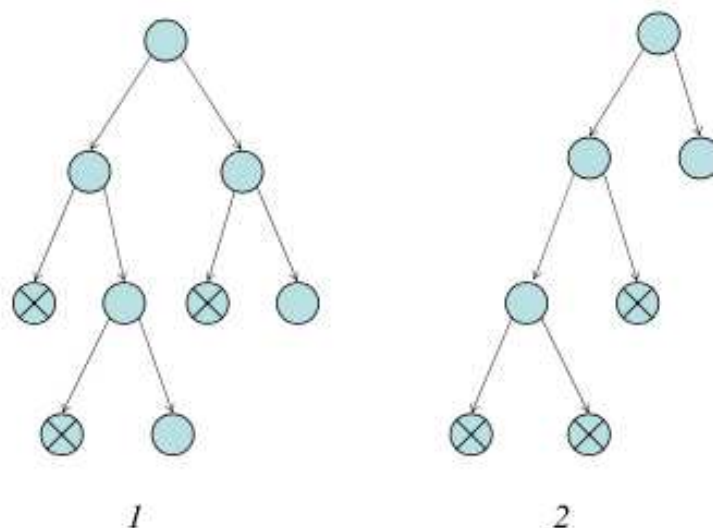


Рисунок 16 – Примеры одновременной (1) и односторонней (2) схем ветвления

В схеме одностороннего ветвления выбирается первая (левая) вершина на нижнем уровне, а для схемы одновременного ветвления такой вершиной может быть любая. Алгоритм заканчивает работу, если зачеркнуты все висячие вершины дерева ветвлений.

Графическое представление метода ветвей и границ иллюстрирует его суть – отсечение ветвей дерева поиска, которое осуществляется на основании сравнения нижней границы и значения функционала на рекорде. Это объясняет название метода [4].

## 1.5 Базы данных и базы знаний

Современные информационные системы, основанные на концепции интеграции данных, характеризуются огромными объемами хранимых данных, сложной организацией, необходимостью удовлетворять разнообразные требования многочисленных пользователей. Основу любой информационной системы составляет база данных, т.е. набор данных, организованных специальным образом. В действующем в настоящее время Законе РФ «О правовой охране программ для электронных вычислительных машин и баз

данных» от 23.09.1992 №3523-1 (ред. От 02.02.2006) дается следующее определение: «База данных – это объективная форма представления и организации совокупности данных (например статей, расчетов), систематизированных таким образом, чтобы эти данные могли быть найдены и обработаны при помощи ЭВМ».

Рассмотрим еще несколько определений термина «база данных».

База данных – это именованная совокупность данных, отражающая состояние объектов и их отношений в предметной области.

База данных – это набор постоянно хранимых данных, используемых прикладными программными системами предприятия.

База данных – это реализация схемы и модели данных на физическом уровне.

База данных – это объект управления в банке данных. База данных описывает состояние объектов предметной области на определенный момент времени совокупностью предложений на некотором формализованном языке.

База данных – это совокупность хранящихся вместе данных при наличии такой минимальной избыточности, которая допускает их использование оптимальным образом несколькими приложениями.

База данных – это совокупность материалов (статей, расчетов, нормативных актов, судебных решений и иных подобных материалов), систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью компьютера.

Все эти определения не являются противоречивыми или взаимоисключающими. Скорее, они представляют разные точки зрения авторов на одно и то же понятие. Сложность определения заключается в том, что компьютерные базы данных за свою не очень длинную историю прошли несколько этапов развития, от файловых систем, хранящих в себе неупорядоченные данные, до постреляционных СУБД, содержимым которых являются данные, обладающие поведением (объекты). Остановимся на еще одном определении.

База данных – это информационная модель предметной области в виде совокупности данных, хранимых в памяти компьютера и связанных между собой по правилам, которые определяют их общие принципы описания, хранения и манипулирования. Под информационной моделью понимают информацию об объекте, отобранную и структурированную в соответствии с заданной целью. Модель данных – это описание методов представления и обработки данных в системе управления базами данных, в том числе методов определения типов и логических структур в базе данных, методов манипулирования данными и методов определения и поддержки целостности базы данных.

Исторически первые базы данных создавались на основе файловых систем, и вся ответственность за работу с ними возлагалась на прикладное программное обеспечение, использовавшее эти базы. Файловые базы данных сегодня практически не применяются. В современной технологии баз данных предполагается, что создание базы данных, ее поддержка и обеспечение доступа пользователей к ней осуществляются централизованно с помощью специального программного инструментария — системы управления базами данных.

Система управления базами данных (СУБД) — это комплекс языковых и программных средств, предназначенный для создания, ведения и применения базы данных пользователями.

Информация, с которой имеют дело информационные системы, разделяется на процедурную и декларативную. Процедурная информация овеществлена в программах, которые выполняются в процессе решения задач, декларативная информация – в данных, с которыми эти программы работают.

Параллельно с развитием структуры компьютеров происходило развитие информационных структур для представления данных. Появились способы описания данных в виде векторов и матриц, возникли списочные структуры, иерархические структуры. В настоящее время в языках программирования высокого уровня используются абстрактные типы данных, структура которых задается программистом.

По мере развития исследований в области информационных систем возникла концепция знаний, которые объединили в себе многие черты процедурной и декларативной информации. В компьютерах знания так же, как и данные, отображаются в знаковой форме – в виде формул, текста, файлов, информационных массивов и т.п. Поэтому можно сказать, что знания – это особым образом организованные данные.

База знаний (БЗ) в информатике и исследованиях искусственного интеллекта – это особого рода база данных, разработанная для оперирования знаниями (метаданными). База знаний содержит структурированную информацию, покрывающую некоторую область знаний.

Полноценные базы знаний содержат в себе не только фактическую информацию, но и правила вывода, допускающие автоматические умозаключения о вновь вводимых фактах и, как следствие, осмысленную обработку информации. Область наук об искусственном интеллекте, изучающая базы знаний и методы работы со знаниями, называется инженерией знаний.

Иерархический способ представления в базе знаний набора понятий и их отношений называется онтологией. Онтологию некоторой области знаний вместе со сведениями о свойствах конкретных объектов также можно назвать базой знаний.

База знаний – семантическая модель, описывающая предметную область и позволяющая отвечать на такие вопросы из этой предметной области, ответы на которые в явном виде не присутствуют в базе. База знаний является основным компонентом интеллектуальных и экспертных систем.

Простые базы знаний могут использоваться для создания экспертных систем и хранения данных об организации: документации, руководств, статей технического обеспечения. Главная цель создания таких баз – помочь менее опытным людям найти существующее описание способа решения какой-либо проблемы предметной области.

Отличия баз знаний от баз данных:

1) Базы данных:

- могут работать с однородными данными;
- представляет собой жестко структурированную модель;
- данные представлены в виде набора записей.

2) Базы знаний:

- могут содержать разнородные и разнотипные данные;
- представляют собой открытую модель;
- знания представлены в виде семантической сети.



## **2 Разработка программного модуля по сборке ПК из комплектующих**

### **2.1 Модель программной системы по сборке ПК из комплектующих**

Модуль по сборке ПК из комплектующих можно разделить на три основных составляющих:

- универсальный сборщик изделий;
- сборщик ПК из комплектующих;
- база данных.

Универсальный сборщик изделий выполняет основную функцию – формирует сборку из комплектующих по заданным параметрам. Включает в себя визуальный редактор, который позволяет просматривать и изменять имеющиеся сборки и комплектующие, а также редактировать существующую базу знаний. Выполняет функцию сервера, который по запросу передает данные о комплектующих или сборках. Также позволяет загрузить в базу данных комплектующие, полученные с помощью синтаксического анализатора.

Сборщик ПК из комплектующих взаимодействует с универсальным сборщиком изделий, передавая запросы от конечного пользователя. Получает ответы сервера (универсальный сборщик изделий) и выводит результаты. Предоставляет пользователю возможность настройки сборки и установки необходимых фильтров для формирования точных запросов.

База данных является главным и единственным хранилищем для всей системы. Все комплектующие, сборки и базы знаний хранятся в базе данных. Возможность подключения к БД имеет только универсальный сборщик изделий.

Также имеется синтаксический анализатор, необходимый для пополнения базы комплектующих. Его ключевой функцией является поиск комплектующих и сохранение данных в определенную структуру, чтобы впоследствии загрузить полученные комплектующие в базу данных, используя универсальный сборщик изделий.

Ниже на рисунке 17 представлена схема программного модуля по сборке ПК.

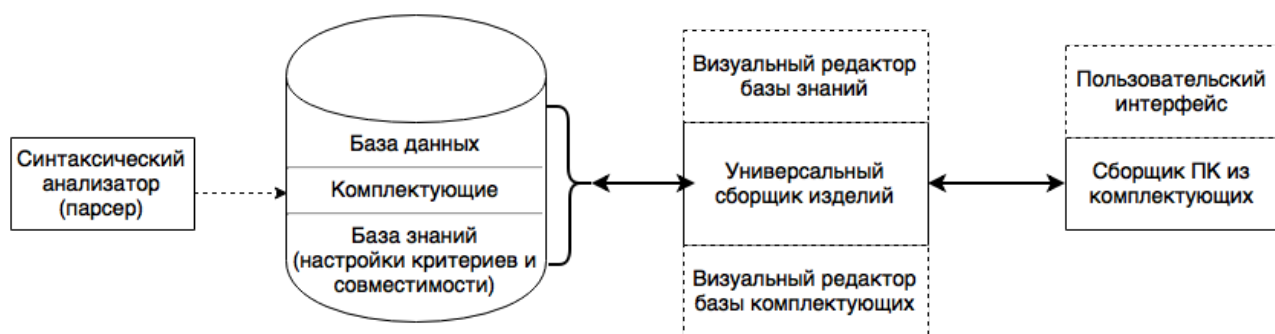


Рисунок 17 – Схема программного модуля по сборке ПК

## 2.2 Средства разработки программного модуля

### 2.2.1 PostgreSQL 9.5

Кроссплатформенная система управления базами данных (СУБД).

#### 2.2.1.1 Руководство по настройке

Для установки программы в операционной системе семейства \*nix используется терминал. Чтобы непосредственно установить PostgreSQL, нужно выполнить следующую команду в терминале:

```
sudo apt-get install -y postgresql-9.5
```

Необходимо создать базу данных и пользователя. Для этого нужно зайти в оболочку СУБД:

```
sudo -u postgres psql
```

Выполнить команды:

```
CREATE DATABASE test_database;
```

```
CREATE USER test_user WITH password 'qwerty';
```

```
GRANT ALL ON DATABASE test_database TO test_user;
```

### **2.2.1.2 Требования по настройке**

PostgreSQL работает на аппаратных платформах x86 и x86-64 на Windows. На других операционных системах работает на таких платформах, как PowerPC, PowerPC64, SPARC, SPARC64, ARMv8-A (64-bit) и предыдущие версии ARM (32-bit, включая такие как ARMv6 на Raspberry Pi), MIPS, MIPSel и PA-RISC.

PostgreSQL доступен на следующих операционных системах: Linux, Windows (Windows 2000 SP4 и позднее), FreeBSD, OpenBSD, NetBSD, OS X (macOS), AIX, HP-UX, Solaris и InnoWare.

Может использоваться даже не очень мощное оборудование, особенно под Linux. И, как в любой СУБД, на производительность влияют: количество памяти, скорость работы дисковой подсистемы и т.д. Рекомендации для выбора аппаратного обеспечения зависят от требования к системе, прогнозируемого размера базы данных, количества пользователей и т. п. Допустимо начинать с минимальной конфигурации, расширяя её по мере надобности.

Переход от одной ОС к другой весьма прост — достаточно сделать резервную копию базы в переносимом формате в одной системе и восстановить на другой.

## **2.2.2 Django 1.11**

### **2.2.2.1 Краткая характеристика**

Свободный фреймворк для веб-приложений на языке Python, использующий шаблон проектирования MVC.

#### **2.2.2.2 Руководство по настройке**

Установка фреймворка происходит с помощью пакетного менеджера pip. Команда установки:

```
pip install django
```

## **2.2.3 JetBrains PyCharm**

### **2.2.3.1 Краткая характеристика**

Интегрированная среда разработки для языка программирования Python от компании JetBrains, разработанная на основе платформы IntelliJ IDEA.

### **2.2.3.2 Руководство по настройке**

Для работы среды разработки необходимо установить JRE (Java Runtime Environment).

### **2.2.3.3 Требования по настройке**

Среда разработки работает под операционными системами Windows, Linux и OS X.

## **2.3 Формирование базы данных**

Вся информация о комплектующих, сборках и правилах сборок хранятся в базе данных. Рассмотрим ее структуру.

Модуль использует 9 таблиц: Product, Services, Users, Product\_Builds, Build, ServiceSettings, Component, ComponentSettings, Actions.

Таблица Product (свойства представлены в таблице 1) содержит все комплектующие и их спецификации. Характеристики комплектующих хранятся в формате JSON для обеспечения универсальности. Это позволяет хранить комплектующие любой предметной области в одной таблице.

Таблица 1 – Свойства таблицы Product

Свойство	Тип	Описание
id	serial	Идентификатор записи в базе данных
specs	jsonb	Характеристики комплектующего
service	integer	Идентификатор сервиса

Таблица Services (свойства представлены в таблице 2) содержит все наименования сервисов каждого пользователя.

Таблица 2 – Свойства таблицы Services

Свойство	Тип	Описание
id	serial	Идентификатор записи в базе данных
name	varchar	Наименование сервиса
user	integer	Идентификатор пользователя

Таблица Users (свойства представлены в таблице 3) содержит все данные о пользователях.

Таблица 3 – Свойства таблицы Users

Свойство	Тип	Описание
id	serial	Идентификатор записи в базе данных
login	varchar	Идентификатор входа в учетную запись
password	varchar	Пароль в зашифрованном виде

Таблица Build (свойства представлены в таблице 4) содержит данные о связи сборок и сервисов.

Таблица 4 – Свойства таблицы Build

Свойство	Тип	Описание
id	serial	Идентификатор записи в базе данных
service_id	integer	Идентификатор сервиса

Таблица Products\_Builds (свойства представлены в таблице 5) содержит данные о связи комплектующих и сборок.

Таблица 5 – Свойства таблицы Products\_Builds

Свойство	Тип	Описание
id	serial	Идентификатор записи в базе данных
product_id	integer	Идентификатор комплектующего
build_id	integer	Идентификатор сборки

Таблица ServiceSettings (свойства представлены в таблице 6) содержит данные о связи настроек сборки изделий и сервисов.

Таблица 6 – Свойства таблицы ServiceSettings

Свойство	Тип	Описание
id	serial	Идентификатор записи в базе данных
service_id	integer	Идентификатор сервиса

Таблица Component (свойства представлены в таблице 7) содержит данные о компонентах сборки и связи с настройками сервисов.

Таблица 7 – Свойства таблицы Component

Свойство	Тип	Описание
id	serial	Идентификатор записи в базе данных
name	varchar	Наименование компонента

Окончание таблицы 7

servicesettings_id	integer	Идентификатор настроек сервиса
--------------------	---------	--------------------------------

Таблица ComponentSettings (свойства представлены в таблице 8) содержит данные о настройках компонента сборки и необходимые условия, по которым происходит сборка изделия.

Таблица 8 – Свойства таблицы ComponentSettings

Свойство	Тип	Описание
id	serial	Идентификатор записи в базе данных
component_id	integer	Идентификатор компонента
key	varchar	Наименование характеристики
action_id	integer	Идентификатор действия
value	varchar	Значение условия

Таблица Actions (свойства представлены в таблице 9) содержит данные о действиях, необходимых для установления условий подбора компонентов.

Таблица 9 – Свойства таблицы Actions

Свойство	Тип	Описание
id	serial	Идентификатор записи в базе данных
action	varchar	Действие для условий подбора компонентов

На рисунке 18 изображена общая структура базы данных.

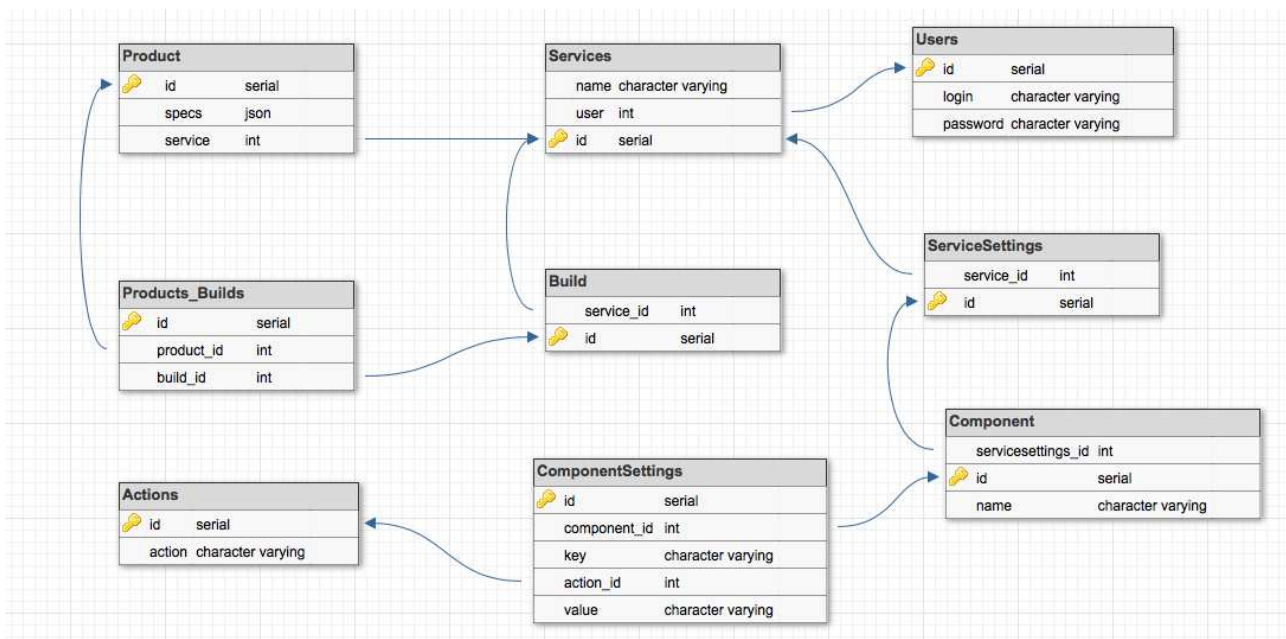


Рисунок 18 – Общая структура базы данных

Ниже представлена часть SQL-кода генерации вышеописанной структуры базы данных.

```

CREATE TABLE «Product» (
    «id» serial NOT NULL,
    «specs» json NOT NULL,
    «service» int NOT NULL,
    CONSTRAINT Product_pk PRIMARY KEY («id»)
) WITH (
    OIDS=FALSE
);

CREATE TABLE «Users» (
    «id» serial NOT NULL,
    «login» character varying NOT NULL UNIQUE,
    «password» character varying NOT NULL,
    CONSTRAINT Users_pk PRIMARY KEY («id»)
) WITH (

```



```

        OIDS=FALSE
    );
CREATE TABLE «Component» (
    «servicesettings_id» int NOT NULL,
    «id» serial NOT NULL,
    «name» character varying NOT NULL,
    CONSTRAINT Component_pk PRIMARY KEY («id»)
) WITH (
    OIDS=FALSE
);
CREATE TABLE «Actions» (
    «id» serial NOT NULL,
    «action» character varying NOT NULL UNIQUE,
    CONSTRAINT Actions_pk PRIMARY KEY («id»)
) WITH (
    OIDS=FALSE
);
CREATE TABLE «ComponentSettings» (
    «id» serial NOT NULL,
    «component_id» int NOT NULL,
    «key» character varying NOT NULL,
    «action_id» int NOT NULL,
    «value» character varying NOT NULL,
    CONSTRAINT ComponentSettings_pk PRIMARY KEY («id»)
) WITH (
    OIDS=FALSE
);
ALTER TABLE «Product» ADD CONSTRAINT «Product_fk0» FOREIGN KEY
(«service») REFERENCES «Services»(«id»);

```

```
ALTER TABLE «Services» ADD CONSTRAINT «Services_fk0» FOREIGN KEY
(«user») REFERENCES «Users»(«id»);
ALTER TABLE «Products_Builds» ADD CONSTRAINT «Products_Builds_fk0»
FOREIGN KEY («product_id») REFERENCES «Product»(«id»);
ALTER TABLE «Products_Builds» ADD CONSTRAINT «Products_Builds_fk1»
FOREIGN KEY («build_id») REFERENCES «Build»(«id»);
ALTER TABLE «Build» ADD CONSTRAINT «Build_fk0» FOREIGN KEY
(«service_id») REFERENCES «Services»(«id»);
```

## 2.4 Разработка алгоритмов работы программной системы

Для работы программной среды по сборке изделия из комплектующих, был разработан универсальный сборщик. Задачей сборщика является формирование сборки на основе заданных пользователем ограничений.

В общем случае поставленную задачу необходимо решать методом ветвей и границ. Предлагается три подхода, позволяющих значительно сократить количество вычислений.

- 1) Полный перебор подразумевает формирование сборки, а затем проверку комплектующих на совместимость. В данной работе проверка совместимости происходит при добавлении каждой новой детали к набору, что сокращает количество необходимых проверок.
- 2) При добавлении новой детали производятся вычисления связанные с проверкой совместимости этой детали, и оценкой её характеристик (цена и т.д.). Количество этих проверок можно уменьшить, если установить порядок выбора типа компонентов согласно количеству таких правил.
- 3) С увеличением числа комплектующих и позиций сборки, вычислительная сложность растет как  $m^n$ . Где  $m$  – среднее количество деталей одного типа, а  $n$  – количество типов. Из всех доступных деталей для каждого типа возьмем только  $M$  «лучших» в смысле критерия подбора (наибольшая

доступная цена и т.д.). В случае, если решение не устраивает пользователя, им вводятся дополнительные фильтры на детали.

Принцип работы сборщика заключается в следующем. На первом этапе происходит полное рассмотрение позиции сборки. То есть, сборщик проверяет какие типы компонентов и в каком количестве необходимы для формирования полноценного изделия в соответствии с заданными настройками. После того как данный пункт проверен и удовлетворяет условиям сборки изделия, сборщик фиксирует позицию и продолжает работу.

На следующем этапе сборщик выбирает деталь, проходя три этапа:

- 1) Удовлетворяет ли выбранный компонент введенным пользователем ограничениям;
- 2) Подходит ли данный компонент к сборке (проверка с помощью заданных настроек на совместимость компонентов);
- 3) Удовлетворяет ли выбранный компонент итоговым характеристикам сборки.

Если хотя бы один из данных пунктов не выполняется, то есть подходящего компонента не было найдено, сборщик возвращается на предыдущую позицию сборки.

В случае, когда все этапы удовлетворяют выбранному компоненту и всей сборке целиком, сборщик переходит к проверке следующего компонента изделия.

После того как все компоненты сборки были отобраны сборщиком, происходит формирование (заполнение) сборки. Затем пользователю выдается результат готового изделия, полностью отвечающего введенным им требованиям и ограничениям.

На рисунке 19 представлена диаграмма вариантов использования.



Рисунок 19 – Диаграмма вариантов использования

Лицо принимающее решение вводит необходимые параметры, которым должен соответствовать собираемый ПК. Впоследствии эти данные передаются посредством GET-запроса на сервер, где они обрабатываются и вызывают функцию по сборке ПК. Далее формируется ответ с готовой сборкой, если таковая собралась, и происходит отправка ответа лицу принимающему решение.

На рисунке 20 представлен алгоритм работы программы, когда используется GET-запрос.

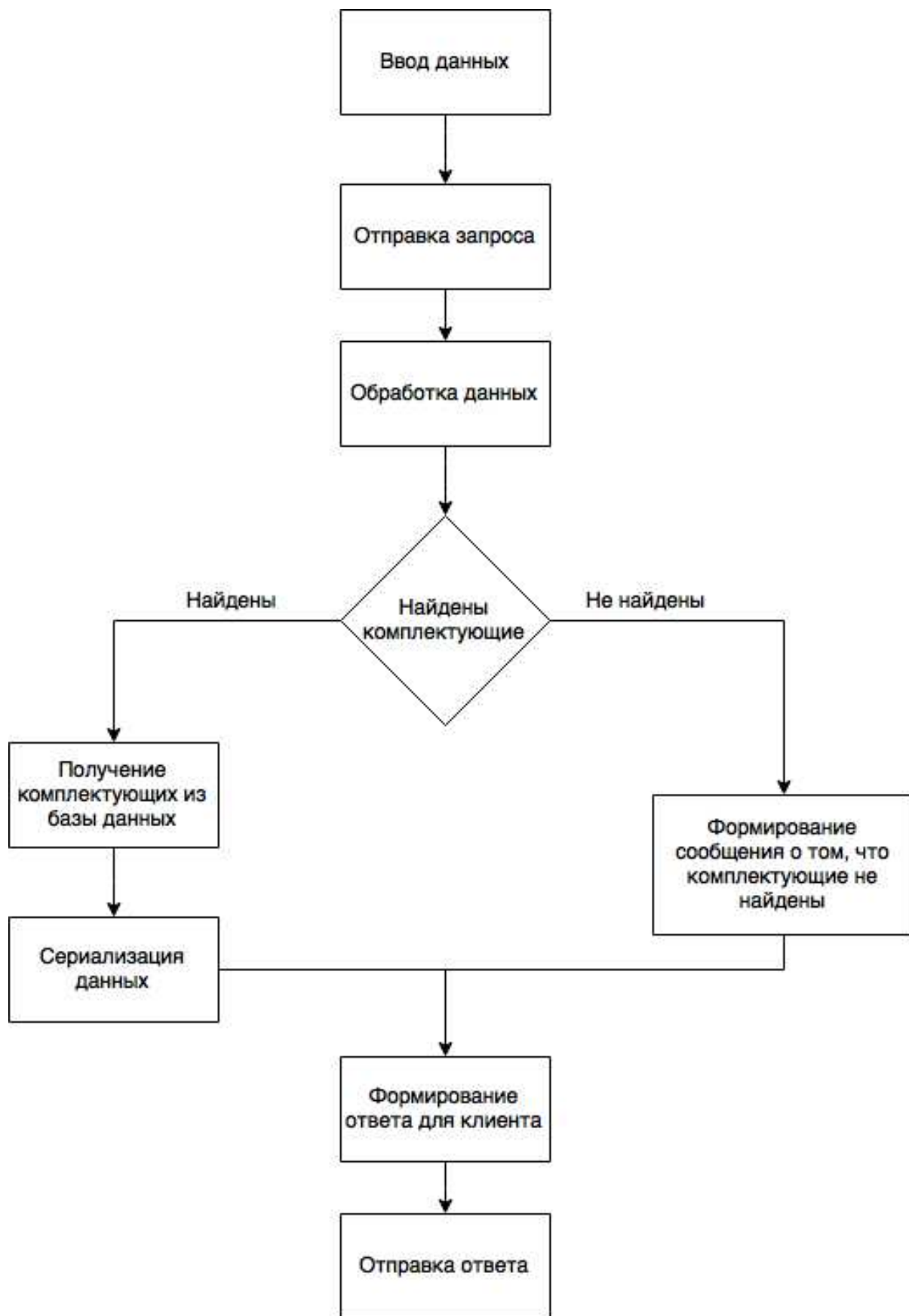


Рисунок 20 – Алгоритм работы модуля при запросе комплектующих

После ввода данных пользователь нажимает на кнопку отправки данных. Исходя из введенных параметров формируется запрос, который отсылается на сервер по адресу /products. Например, если нужно вывести товары категории «Процессор» стоимостью от 10000 рублей до 20000 рублей, то формируется такой запрос:

[http://www.diplom.ru/products?category=процессор&min\\_price=10000&max\\_price=20000](http://www.diplom.ru/products?category=процессор&min_price=10000&max_price=20000)

Метод GET отправляет собранную информацию как часть URL, поэтому эти данные присутствуют в адресе.

Если после получения запроса комплектующие были найдены, то происходит формирование списка запрошенных комплектующих и отправка ответа пользователю.

На рисунке 21 представлен алгоритм работы программы при сборке изделия.

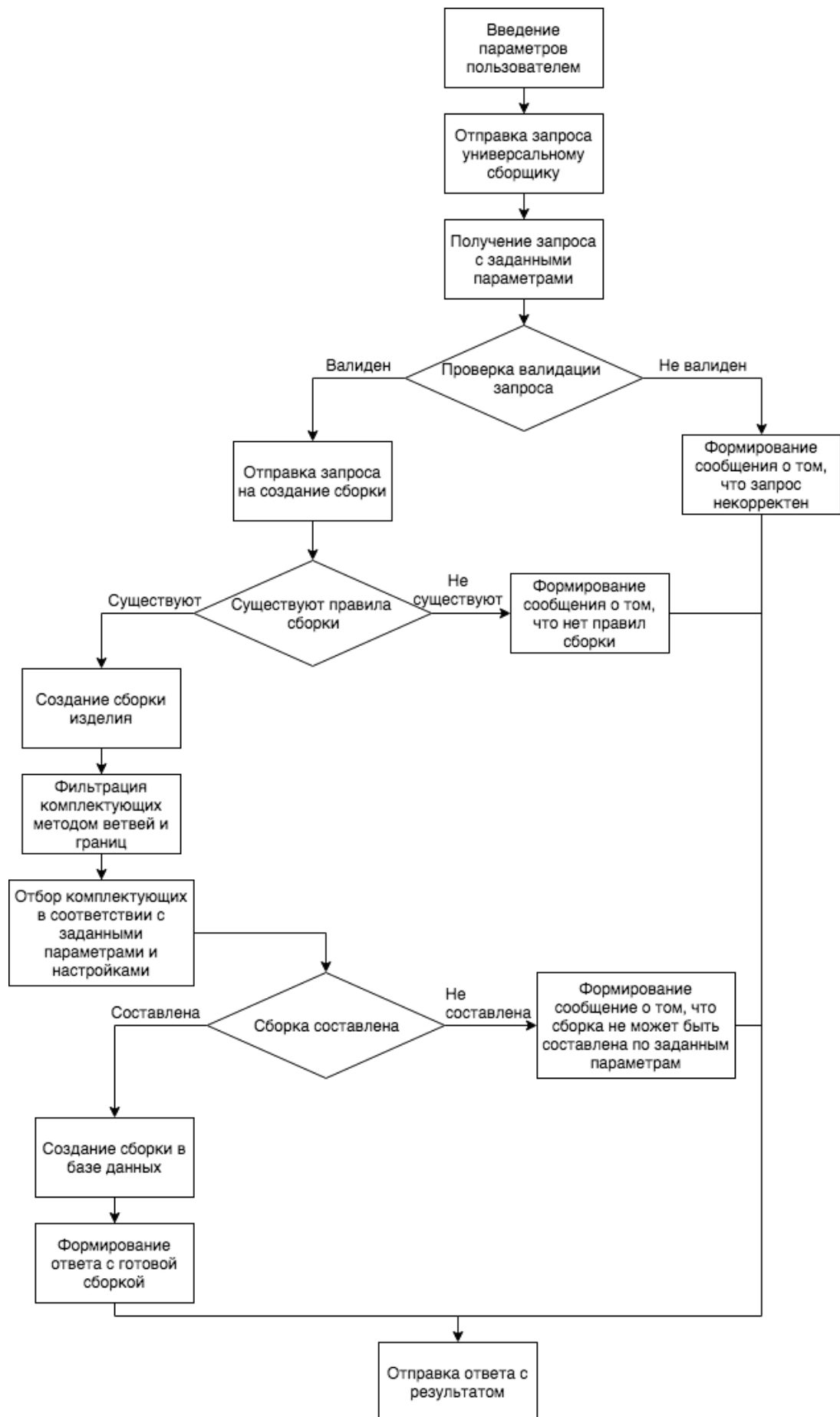


Рисунок 21 – Алгоритм работы модуля при сборке изделия

Введенные пользователем данные отправляются на сервер, где происходит валидация запроса. Если запрос не проходит валидацию, то формируется сообщение об ошибке, иначе отправляется запрос на создание сборки. В случае, если правила сборки не созданы администратором, то происходит формирование сообщения об ошибке. Если же правила сборки существуют, то выполняется процедура сборки изделия. Первым делом происходит фильтрация компонентов и отбор комплектующих по заданным параметрам. Когда сборка готова, то данные об этой сборке сохраняются в базе данных, а пользователю формируется ответ, в котором содержится готовый вариант сборки. Если же сборка не составлена, то формируется сообщение о том, что сборка не может быть составлена по заданным параметрам и данное сообщение отправляется пользователю.



## **ЗАКЛЮЧЕНИЕ**

В результате анализа существующих сайтов интернет-магазинов, а так же специализированного ПО, было выявлено, что на данный момент не существует полноценной программной среды по сборке изделия из комплектующих. Аналоги данных продуктов имеют место в наше время, но они лишь способны помогать пользователю с выбором.

В данной работе были рассмотрены задачи, необходимые для создания программной среды по сборке готового изделия из комплектующих. В ходе работы были выявлены необходимые характеристики комплектующих и их типы.

Разработанная база данных является универсальной, и может быть использована во многих предметных областях. В её состав входит свод таблиц, которые содержат не только характеристики комплектующих, но и экспертную информацию для определения совместимости и расчетных параметров.

Программный модуль, разработанный в данной работе, осуществляет подбор компонентов, основываясь на введенных пользователем характеристиках. Результатом его работы является формирование полноценной сборки, полностью отвечающей требованиям пользователя.

Апробация программного модуля по сборке изделия из комплектующих была произведена на примере персонального компьютера.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Таненбаум, Э. Архитектура компьютера / Э. Таненбаум, Т. Остин. – 6-е. изд. – Санкт-Петербург : Питер, 2017. – 816 с.
- 2 Ватаманюк, А.И. Обслуживание и настройка компьютера / А.И. Ватаманюк. – Санкт-Петербург : Питер, 2009. – 441 с.
- 3 Степаненко, О.С. Практическая сборка и наладка ПК. Самоучитель / О.С. Степаненко. – Москва : ИД Вильямс, 2007. – 336 с.
- 4 Соломенчук, В.Г. Аппаратные средства персональных компьютеров / В.Г. Соломенчук. – Санкт-Петербург : БХВ-Петербург, 2003. – 512 с.
- 5 Гончаров, Е.Н. Исследование операций. Примеры и задачи : учебное пособие / Е.Н. Гончаров, А.И. Ерзин, В.В. Залюбовский. – Новосибирск : НГУ, 2005. – 78 с.
- 6 Крёнке, Д. Теория и практика построение баз данных: учебное пособие / Д. Крёнке. – Санкт-Петербург : Питер, 2003. – 800 с.
- 7 Гаврилова, Т.А. Базы знаний интеллектуальных систем: учебник / Т.А. Гаврилова, В.Ф. Хорошевский. – Санкт-Петербург : Питер, 2017. – 384 с.
- 8 Казиев, В.М. Введение в анализ, синтез и моделирование систем / В.М. Казиев. – Москва : Бином. Лаборатория знаний, 2007. – 248 с.
- 9 Хомоненко, А.Д. Базы данных : учебник для высших учебных заведений / А.Д. Хомоненко, М.Г. Мальцев, В.М. Цыганов. – 6-е изд. – Санкт-Петербург : Корона-Принт, 2009. – 736 с.
- 10 Кузин, А.В. Базы данных: учебное пособие / А.В. Кузин, С.В. Левонисова. – 5-е изд. – Москва : Академия, 2012. – 320 с.
- 11 Анализ данных и процессов : учебное пособие / А.А. Барсегян, М.С. Куприянов, И.И. Холод, М.Д. Тесс, С.И. Елизаров. – 3-е изд., перераб. и доп. – Санкт-Петербург : БХВ-Петербург, 2009. – 512 с.
- 12 Левитин, А. Алгоритмы. Введение в разработку и анализ / А. Левитин. – Москва : Вильямс, 2003. – 576 с.

- 13 Новосельцев, В.И. Теоретические основы системного анализа / В.И. Новосельцев, Б.В. Тарасов.— Москва : Майор, 2013. — 536 с.
- 14 Алгоритмы: Построение и анализ / Т.Х. Кормен, Ч.И. Лейзерсон, Р.Л. Ривест, К. Штайн. — 2-е изд.— Москва : Вильямс, 2012. — 1296 с.
- 15 ECMA-404 The JSON Data Interchange Format. — Введ. 10.2013. — Geneva : Ecma International, 2013. — 8 с.
- 16 Django documentation [Электронный ресурс] // Сайт. — Режим доступа: <https://docs.djangoproject.com/en/dev/>
- 17 Django Rest Framework [Электронный ресурс] // Сайт. — Режим доступа: <http://www.django-rest-framework.org/>



Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
ИНСТИТУТ КОСМИЧЕСКИХ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ  
Кафедра «Информатика»

УТВЕРЖДАЮ

Заведующий кафедрой

 А. С. Кузнецов

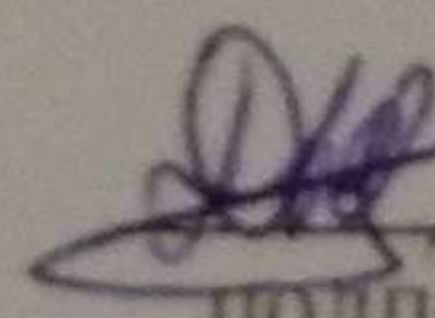
« 19 » 06 2017 г.

**БАКАЛАВРСКАЯ РАБОТА**

27.03.03 «Системный анализ и управление»

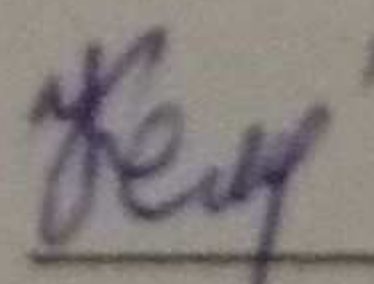
Разработка экспертного модуля по сборке изделия из комплектующих

Руководитель

 19.06.17  
подпись, дата

доцент, канд. техн. наук А.А. Даничев

Выпускник

 19.06.17  
подпись, дата

Е.С. Кемеров